CS 790-02: Optimizing Compilers for Modern Architectures

Meilin Liu
Wright State University - Main Campus, meilin.liu@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
Course Description

This course studies compiler optimization for modern architectures. Between parsing the input program and generating the target machine code, optimizing compilers perform a wide range of program transformations on a program to improve its performance. In this course we focus on data dependence analysis, loop transformations, loop scheduling, cache management, and a combination of these optimizing techniques.

Lecturer
Meilin Liu
Office: 353 Russ Engineering Center
Phone: 937-775-5061
Office Hours: 1:45 – 2:45 pm Monday/Wednesday; 5:00 – 6:00pm Tuesday/Thursday
Email: meilin.liu@wright.edu
Web: www.wright.edu/~meilin.liu

Class
• Monday/Wednesday 8:00–9:15 pm Russ Engineer Cntr 208

Text

Reference
*High Performance Compilers for Parallel Computing, Michael Wolfe, Addison-Wesley, 1996
*Parallel Computer Architecture, David Culler and Jaswinder Pal Singh, Morgan Kaufmann, 1999

Required Work
Homework 30%
Project 40%
Final Exam 30%

Grading
The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

Policies and Notes
• Attendance: Attendance is not required, but maybe documented by the pop up quizzes. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions
will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!

- I will utilize webCT (wisdom.wright.edu) to post updates to the course, solutions, assignments, announcements, schedule, etc. Get in the habit of checking it regularly.

- Always make back ups of all of you work. Never have just one copy of anything!

- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.

- If you need to meet me other than my office hours, better make an appointment by email or by phone beforehand.

**Academic Misconduct**

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to write the assignment, complete and submit your own work.

Work that has obviously been copied or in the more extreme case, when the original author's name has not even been changed, both parties will receive a 0 grade for that assignment. Both parties will also be turned over to the Office of Judicial Affairs.

### Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Contents</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Architecture Overview</td>
<td>Chap. 1</td>
</tr>
<tr>
<td>2</td>
<td>Data Dependence Theory</td>
<td>Chap. 2</td>
</tr>
<tr>
<td>3</td>
<td>Data Dependence Analysis</td>
<td>Chap. 3</td>
</tr>
<tr>
<td>4</td>
<td>Preliminary Transformations</td>
<td>Chap. 3, 4</td>
</tr>
<tr>
<td>5</td>
<td>Fine grained parallelization</td>
<td>Chap. 5</td>
</tr>
<tr>
<td>6</td>
<td>Scheduling</td>
<td>Chap. 10</td>
</tr>
<tr>
<td>7</td>
<td>Control dependence</td>
<td>Chap. 7</td>
</tr>
<tr>
<td>8</td>
<td>Register management</td>
<td>Chap. 8</td>
</tr>
<tr>
<td>9</td>
<td>Cache management</td>
<td>Chap. 9</td>
</tr>
<tr>
<td>10</td>
<td>Cache management, Review</td>
<td>Chap. 9</td>
</tr>
</tbody>
</table>